8. **Construct a program to simulate the Least Recently Used paging technique of memory management. When a page must be replaced, the oldest page is chosen**

#include<stdio.h>

int findLRU(int time[], int n){int i, minimum = time[0], pos = 0;for(i = 1; i < n; ++i){if(time[i] < minimum){minimum = time[i];pos = i;

}}

return pos;

}

int main()

{int no\_of\_frames, no\_of\_pages, frames[10], pages[30], counter = 0, time[10], flag1, flag2, i, j,pos, faults = 0;printf("Enter number of frames: ");scanf("%d", &no\_of\_frames);printf("Enter number of pages: ");scanf("%d", &no\_of\_pages);printf("Enter reference string: ");

for(i = 0; i < no\_of\_pages; ++i){scanf("%d", &pages[i]);

}for(i = 0; i < no\_of\_frames; ++i){frames[i] = -1;

}for(i = 0; i < no\_of\_pages; ++i){

flag1 = flag2 = 0;for(j = 0; j < no\_of\_frames; ++j){if(frames[j] == pages[i]){counter++;time[j] = counter;flag1 = flag2 = 1;

break;

}

}if(flag1 == 0){for(j = 0; j < no\_of\_frames; ++j){ if(frames[j] == -1){ counter++; faults++; frames[j] = pages[i]; time[j] = counter; flag2 = 1; break;

}}}if(flag2 == 0){ pos = findLRU(time, no\_of\_frames); counter++; faults++; frames[pos] = pages[i];

time[pos] = counter;

} printf("\n"); for(j = 0; j < no\_of\_frames; ++j){

printf("%d\t", frames[j]);

}}

printf("\n\nTotal Page Faults = %d", faults); return 0;

}

OUTPUT:

